# An Automatic Generation of Production Documentation from MultiProLan Models

Milica Todorović*, Đorđije Ivanišević*, Marko Vještica*, Vladimir Dimitrieski*, Ivan Luković**
*University of Novi Sad, Faculty of Technical Sciences, 21000 Novi Sad, Serbia
** University of Belgrade, Faculty of Organizational Sciences, 11000 Belgrade, Serbia
milicatodorovic@uns.ac.rs, djordjij3_ivanis@uns.ac.rs, marko.vjestica@uns.ac.rs, dimitrieski@uns.ac.rs, ivan.lukovic@fon.bg.ac.rs

***Abstract*—In the traditional manufacturing industry, technical documentation comes in various forms and plays an important role at all stages of a product development and production. In recent years, the idea of Industry 4.0 has been popularized. One of the core elements of Industry 4.0 is a digital twin representing a virtual model of physical objects and processes in a factory. An initial version of the digital twin is based on the documentation, and all changes that later occur in the lifetime of the digital twin must also be reflected in the corresponding documents. Current forms of the documentation are unsuitable for both the efficient creation of the initial digital twin version, and for consequent updates. To bridge this gap, a novel approach is proposed in this paper. The core element of this approach is a domain-specific modeling language – MultiProLan. Its models serve as centralized representation of knowledge about the production processes. Different types of technical documentation can be automatically generated from the models, resulting in increased document consistency, quality and minimized number of errors introduced by humans. We also explore the possibility of automatic generation of different types of technical documentation from such models, as well as different production process modeling aspects that must be covered.**

## I. Introduction

In traditional manufacturing, technical documentation has an important role in production process development, planning and realization. To describe different aspects of production processes, companies utilize various forms of technical documentation, including different schemas, drawings and textual documents. These various forms are often either legally required or prescribed by a standard. For example, Bill of Materials (BOM) [1] is often used to describe which materials are needed for a product assembly. BOM is required by multiple legal regulations including the European Union's Restriction of Hazardous Substances (RoHS) directive [2] and California's Proposition 65 law [3], as well as by multiple standards from the ISO Standards catalogue (ICS) 01.110 [4]. Similarly, Flow Process Chart (FPC), which is used to represent a workflow of production processes, is a standard published by American Society of Mechanical Engineers (ASME) [5]. Often, in order to get a full overview of a whole production process, multiple documents conforming to different standards and in different formats must be taken into consideration [6].

Once created, the entire technical documentation must be maintained during a production process lifecycle. Changes to the production process are inevitable, whether they are a consequence of the processes optimization, production volume change or production system change. Whenever a change occurs, it must be properly documented. These modifications are often added as separate documents, increasing the volume of documentation needed to get an overall picture of production processes. In the worst case, these modifications are not documented at all, therefore causing a drop in an overall quality of technical documentation [7].

In recent years, the idea of Industry 4.0 (I40) was born and is supported by multiple governments. I40 should utilize new technologies in order to ensure a better flexibility and scalability of manufacturing systems [8]. I40 aims to provide and enhance transparency in production systems and allow real-time production control through the use of Cyber-Physical Production System (CPPS). CPPS consists of autonomous and co-operative elements and subsystems that are connected across all levels of production – from processes through machines up to production and logistic networks. A digital twin represents a virtual model of physical objects and processes in a factory and is a prerequisite for the development of CPPS. I40 puts further emphasis on the digital twin as it's no longer used only for static optimization, but also for dynamic real-time control and optimization [9], [10]. To create an initial version of such digital twin, technical documentation may be utilized. This initial version of a digital twin evolves and changes due to the production process optimization and real-time tracking. Consequently, all these changes must be reflected in the technical documentation to ensure its desired quality [6].

However, technical documentation in its current form is not suitable for simple and efficient creation of a digital twin, as multiple documents with different formats must be parsed. Correspondingly, updating documentation based on a digital twin's changes poses a challenge as multiple documents must be updated and tracked. Changing a documentation format or eliminating it completely in a favor of a digital twin would pose an enormous challenge, as it is both widely used and often legally required.

To bridge this gap, a novel approach for the technical documentation creation and maintenance is required. One such approach, suggested in this paper, is based on the existence of a centralized point of knowledge about production processes suitable both for the creation of a digital twin and the creation of required forms of technical documentation. This knowledge is stored in models created by Multi-level Production process modeling Language (MultiProLan) – a Domain-Specific Modeling Language

 

(DSML) implemented to enable the modeling of production processes suitable for automatic execution [11]. Accordingly, updating a single point of knowledge poses a lesser challenge than updating various documents. Finally, all required forms of documentation can be generated from the central point of knowledge.

Apart from Introduction and Conclusion, this paper is structured as follows. In Section 2 an overview of the related work is given. The proposed architecture is described in Section 3. In Section 4 the extended syntax of MultiProLan and generators are presented.

## II. Related Work

To the best of our knowledge there is a few papers about creation and maintenance of technical documentation in I40. However, none of them focuses on generation of documentation, but rather on detection of events which trigger documentation updates, or on code generation for the execution layer.

Barthelmey et al. [7] have established that a flexible production system, combined together with the rising complexity of machinery, increases the amount of technical documentation required. For manufacturers, this results in both larger cost of an initial creation of documentation and the increased cost and effort required to maintain documentation. To solve this, they propose a methodology for a self-organized creation and maintenance of technical documentation based on networking all the components and modules with Cyber Physical Systems (CPS). CPS is a computer system in which a mechanism – in this case a production plant, is controlled or monitored by computer-based algorithms. For this, a digital twin of all processes, as well as a digital twin of all factory's resources are required. Every event that occurs, i.e., a resource successfully performed a production step, or a resource was modified, is detected by CPS, and sent to a digital twin to update its status. Required documentation is generated from data owned by a digital twin. The focus of their paper is on the possible design of event detection, while the existence of a digital twin and possibility to generate the documentation from a digital twin is assumed as prerequisite.

Lenkenhoff et al. [12] further expand the proposal given by Barthelmey et al. [7] by proposing a similar architecture based on CPS that would enable up-to-date representation of a machinery and factory floors. Likewise, the focus of their paper is also on a design of events and event detection. However, they do suggest that AutomationML [13] – an XML based exchange format often used in industry, can be used to describe machine's components and arrange them hierarchically to describe complete objects. According to them, this description can serve as a base for the creation of a digital twin and can also be easily updated when a machine is changed in any way, i.e., a new part is added.

Weissenberger et al. [14] proposed a model-based code generation approach in order to lower the cost of implementing Manufacturing Execution Systems (MES) in Small and Medium Enterprises (SMEs). To achieve this goal, the authors extended the syntax of MES Modeling Language (MES-ML). MES-ML is a graphical modeling language for an interdisciplinary specification of MES [15]. One of the extensions authors added to MES-ML is meant to enable modeling of the reports displayed to workers. Both structure and content of the reports can be modeled, while the actual data is provided by the MES later on. However, evaluation of the syntax is still in progress and there are no automatic generators for the reports.

In this paper, our focus is on how the documentation can be generated instead of manually written. We propose an architecture where knowledge about a production process is centralized in a form of a model. Documentation is then generated from such a model. In future, this approach can be extended to enable real-time collection of data from the digital twin to finalize the link between the data and the documentation, as proposed in aforementioned papers.

## III. Proposed Architecture

As presented in Section 1, various aspects of production processes are spread across multiple documents. To create a digital twin of a production process, full overview of the process is required. Currently, to gain the overview multiple documents in different formats must be reviewed. Likewise, any changes done to the digital twin and the real process must be mirrored to technical documentation. Both the process of obtaining the full overview required for the digital twin and the process of transferring the changes back to the documentations are cumbersome and error prone. To bridge the gap between a digital twin and current forms of technical documentation, a novel Model-Driven (MD) approach is suggested.

The core element of this approach is a production process model, a single point of knowledge specifying various relevant aspects of a production process. This single point of knowledge can serve as an efficient starting point for creation of a digital twin's initial version. Correspondingly, it can also be updated in a more efficient manner compared to the process of manually updating multiple documents following different standards. Finally, all the required forms of technical documentation can be automatically generated from a production process model. This approach to documentation generation provides the following benefits: (i) easier modification of a document template when required by legal regulations, standards or market changes; (ii) simplified process of keeping documents up to date since all changes are only applied to the single point of knowledge instead of being spread over multiple documents; and (iii) increased document consistency and quality, as well as minimized number of human errors due to the documents being automatically generated in a non-stochastic manner.

To enable these benefits, a created production process model needs to have a sufficient level of details to enable automatic generation of the most frequently used technical documents. Currently, there are multiple well-known languages used to model different processes, including Business Process Model and Notation (BPMN) [16], Unified Modeling Languages (UML) activity diagrams [17] and Petri nets [18]. However, none of them are tailored specifically for the modeling of production processes, leaving them unable to express required level of details [19]. For example, products required for a production step would have to be expressed as annotations if BPMN is used. Similarly, if Petri nets are used it is altogether impossible to define these products. Considering all this, a DSML named MultiProLan [11] was created specifically for the domain of production process modeling. We chose to extend this language with additional modeling concepts to enable the automatic generation of different forms of documentation. The choice to use MultiProLan was based on the in-depth analysis performed by Vještica et al. [19] which compared multiple different languages and their ability to express different aspects of a production process. This language has also been applied in industry small-scale assembly use cases [20].

MultiProLan is a modeling language, created as a part of a larger system [10], meant for automatic production orchestration and process execution. Its models are automatically transformed into executable instructions that are sent to human workers or machines. The language also contains some of concepts from different production process documents such as BOM and ASME FPC. Because of this, models created with MultiProLan should be suitable for the automatic generation of at least said documents. The abstract syntax of MultiProLan was available to the authors, and therefore it was possible to expand this syntax to enable generation of these, as well as of other often used documents. These extensions are explained in more detail in Section 4.

Once a MultiProLan based production process model is created with the expanded syntax, Model-to-Text (M2T) transformations enable the creation of desired forms of technical documentation. Because rules for M2T transformations are defined at the language level document consistency and template compliance are guaranteed each time a document is generated. When required, existing generators can be modified, and new generators can be added in order to support new requests caused by legal regulations, standards or market changes. However, in most cases such changes should not require existing models to be recreated. When required, manual editing of generated documents is also possible.

The simplified system architecture concerning only the automatic generation of technical documentation is displayed in Fig. 1 in dashed rectangle. It is important to note that proposed architecture in no way obstructs the original system MultiProLan was created for, but rather extends it with a possibility to generate required forms of technical documentation. The production process model is created by a process designer using a MultiProLan-based modeling tool. The term process designer denotes a user that represents any other role involved in the production process specification – a plant manager, a process engineer or a quality engineer. The created model is then consumed by code generators. Each code generator is responsible for the creation of a single form of production process documentation which it gives as an output. This allows for a single generator to be easily added, removed or modified, without affecting the other generators, making it simple to customize which documentation is generated. Currently, generators for the following documents are created: ASME FPC [5], BOM [1], Bill of Materials and Operations (BOMO) [1], Job Breakdown Sheet (JBS) [21] and Failure Mode and Effects Analysis (FMEA) [22].

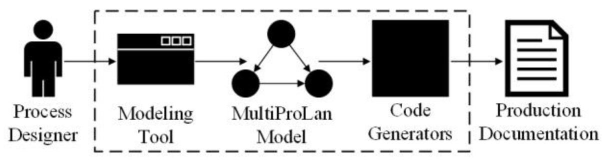


Figure 1. The simplified system architecture for automatic generation of technical documentation

## IV. The Extension Of MultiProLan Abstract Syntax To Support The Generation Of Technical Documentation

As previously mentioned, different documents are used to describe different aspects of production processes. Documents containing different aspects were selected in order to test the possibility of automatic generation from a MultiProLan models. For example, BOM specifies a product structure, while JBS shows process steps in a form suitable for worker training. In addition to showcasing different aspects, selected documents are also either legally required, proposed by a standard, widely used in industry or a combination of that. To begin with, MultiProLan already contained some of the modeling concepts from different production process documents such as BOM and ASME FPC. Missing concepts used in these two documents, as well as concepts required to generate BOMO, JBS and FMEA were added to the MultiProLan syntax.

The expanded syntax of MultiProLan is displayed in Fig. 2. This is only an excerpt of the syntax relevant for this paper, while the rest of it can be found in our previous paper [11]. The remainder of this chapter explains why each document was selected, as well as the motivation behind the additions to the syntax.

All examples of the generated documents displayed in this section were created for the model of Muffin Box Production Process (Fig. 3). The model presented here represents a simplified version of the real production process, due to space limitations. Process steps in the model are abstract and can be further broken down to simpler steps.

To create the final product cupcakes must be baked and appropriate packaging must be prepared. Since these two parts are independent of each other, they can be performed in parallel. To prepare the packaging workers first gather the box (*Get Box* step) and the lining paper (*Get Paper* step). Once both the box and the paper are brought, a packaging box can be assembled (*Place Paper in Box Step*). Simultaneously, muffins are being baked. Firstly, muffin mix is transported from a storage (*Get Muffin Mixes* step). Then, both the chocolate and vanilla muffins are prepared at the same time (*Prepare Choco Muffin* and *Prepare Vanilla Muffin* steps). Before the final assembly, muffins must first cool down (*Wait to Cool Down step*). Once both the box and the muffins are ready, final product is assembled in *Put Muffins* step. Due to space limitations, examples of some generated documents have been not been disclosed in this paper.

BOM specifies the structure of a product in terms of the components, subassemblies and assemblies that constitute the product. BOM is one of the inputs that are required for Material Requirements Planning (MRP) – a method used to solve production and inventory management problems. MRP is prevalent in industry as a basic planning mechanism for many manufacturing companies [23]. The MultiProLan syntax can express process steps (*ProcessStep* class), as well as input and output products of each step (*inProducts* and *outProducts* relationships between classes *ProcessStep* and *Product*). However, the original syntax did not have an option to specify the quantity, which is required for BOM, so attribute *quantity* was added to the *Product* class. The *isFinal* attribute was added to enable

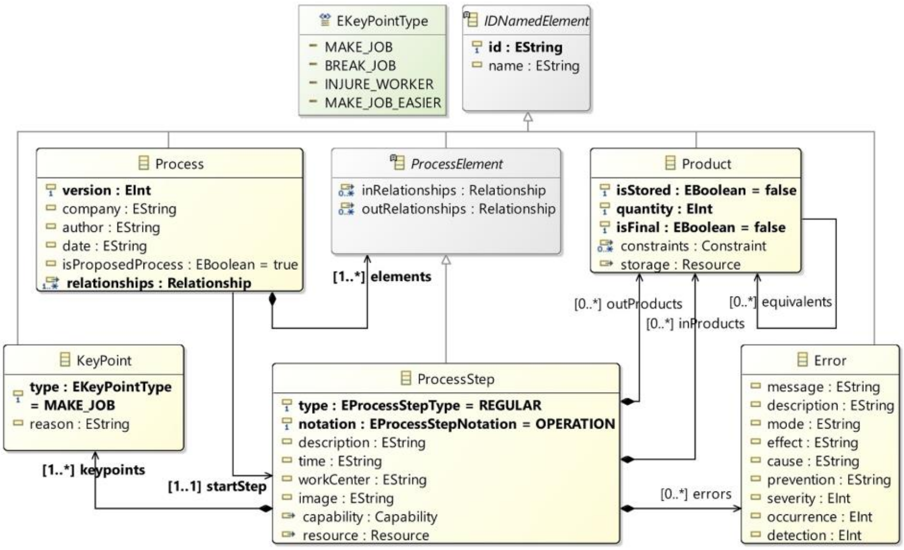


Figure 2. The MultiProLan meta-model extensions

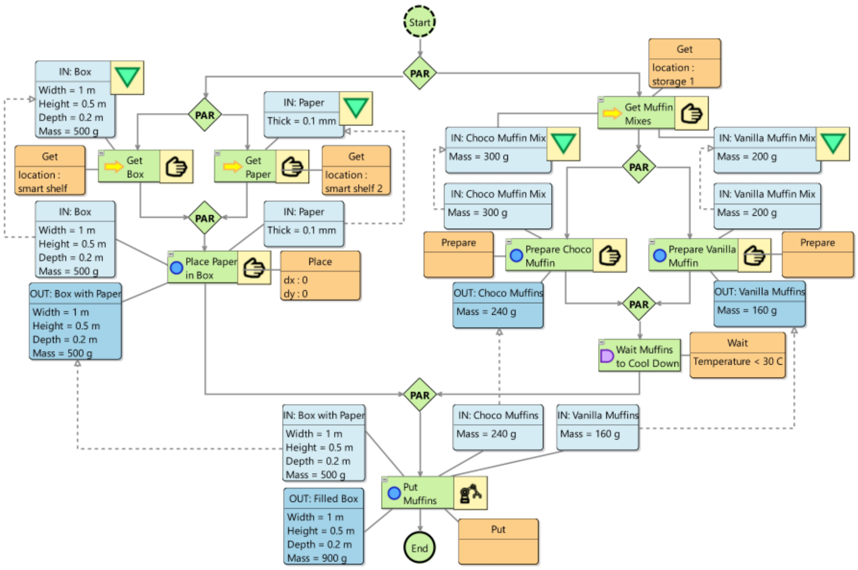


Figure 3. The MultiProLan meta-model extensions

differentiating between a main product of a production. An example of generated BOM is displayed in Fig. 4. In future work, MultiProLan syntax can also be modified to support generation of Generic BOM (GBOM). GBOM is a document which provides the means to easily describe many variants within a product family. MultiProLan already contains the concept of equivalent product. However, syntax would have to be further extended to enable modeling of rules that define which combinations of component variants result with a valid final product.

Flow diagrams are a technique for visual organization and structure of a processes' overview. Flow diagrams are often used for big-picture analysis and optimization of production processes. There are three main types of flow diagrams used to describe production processes: (i) a single object – used for tracking a single product or a single resource through the process; (ii) an assembly/disassembly – used to describe steps needed to assemble or disassemble a product, as well as parts used or produced in each step; and (iii) an action-decision – used for visualization of different process variants [23]. The created generator uses the ASME FPC format to generate a single object flow diagram, tracking the steps for production of the final product. Since flow diagrams are often used for optimization, attributes *time* and *workCenter* were added to the *ProcessStep* class. In Fig 5, we present an example of generated ASME FPC. The MultiProLan syntax can specify decisions and conditions under different steps that are performed with the use of class *Relationship*. Therefore, action-decision flow diagrams can also be generated in the future. Single object diagrams that follow a single resource can also be generated, as the MultiProLan syntax supports the option to assign a resource to a process step.

Finally, as the MultiProLan syntax allows the user to define multiple input and output products for each process step (*inProducts* and *outProducts* relationships between classes *ProcessStep* and *Product*), assembly/disassembly flow diagrams can also be generated. To illustrate this, the BOMO generator is created. BOMO was proposed by Jiao et al. [1] as a data structure that unifies BOM and Bill of Operations (BOO) into a single set. This was done to

| BOM chart | ☑ Proposed Process<br>☐ Present Process | Final Product<br>Filled Box | University of Novi Sad |
|---|---|---|---|
| **Hierarchy Level** | **Product** | **Component** | **Quantity** |
| 1 | Filled Box | Vanilla Muffins | 10 |
| 1 | Filled Box | Box with Paper | 1 |
| 1 | Filled Box | Choco Muffins | 15 |
| .2 | Vanilla Muffins | Vanilla Muffin Mix | 1 |
| .2 | Box with Paper | Paper | 1 |
| .2 | Choco Muffins | Choco Muffin Mix | 1 |
| .2 | Box with Paper | Box | 1 |
| **Author** | | **Date** | |
| Marko Vjestica | | May 30, 2021 | |

Figure 4. Example of generated BOM

synchronize multiple perspectives on variety such as customer ordering, product engineering, and operations planning.

Throughout its lifecycle, a production process will often be evaluated for various reasons. FMEA is a design-evaluation procedure used to identify all conceivable and potential failure modes and determine their effect on system performance. FMEA is used to identify failure situations early in the design cycle before production testing. During FMEA technical documentation is used to: (i) standardize the process; (ii) as a means of historical documentation; and (iii) as a basis for future development [23]. MultiProLan syntax was extended with the class *Error*. Error represents a single possible failure situation recognized during the FMEA process. Multiple failure situations can be defined for a single process step (*errors* relationship between classes *ProductStep* and *Error*).

Another important step during a process lifecycle is operator training. Operators must understand process steps and equipment well to perform their job in an efficient and safe manner. To ensure required efficiency of the workers in the war-related industries during the World War II, United States of America launched the Training Within Industry (TWI) service. TWI service training program introduced Job Breakdown Sheets (JBS), a document used to further break down and analyze single production process step in a way which highlights the most important moments of the step preformation. JBS is still used in industry today by large companies such as Toyota [21]. JBS contains all the steps that an operator must perform. In addition to steps being named and textually described, images are also often used to help with the visualization. Therefore, attribute *image* was added to class *ProcessStep*. For each step multiple key points are also noted. Each key point notes a situation that will either result in step failure or success, put the worker in danger or make the job easier. To enable this in detail description of a production process step following changes were added: (i) the *image* attribute in class *ProcessStep*; (ii) class KeyPoint describing a single key situation; and (iii) association between a *ProcessStep* and related *KeyPoints*.

## V. Conclusion

To cope with challenges of creating and updating technical documentation for production processes, a novel approach was proposed. The idea behind this approach is to create a centralized point of knowledge in the form of a production process model. An implementation of such

| FPC chart | ☑ Proposed Process<br>☐ Present Process | | Final Product<br>Filled Box | | University of Novi Sad | | |
|---|---|---|---|---|---|---|---|
| **Process Step ID** | **Process Step Name** | **Work Center** | Operation | Inspection | Transport | Delay | Storage |
| 10 | Start | | ● | □ | ⇨ | D | ▽ |
| 20 | Parallelism | PREPARE THE BOX AND THE MUFFINS | | | | | |
| 20.10 | Get Muffin Mixes | Storage Room | ○ | □ | ➡ | D | ▼ |
| 20.20 | Parallelism | BAKE MUFFINS | | | | | |
| 20.20.10 | Prepare Choco Muffin | Kitchen | ● | □ | ⇨ | D | ▽ |
| 20.20 | Parallelism | PARALLELISM END | | | | | |
| 20.20.10 | Prepare Vanilla Muffin | Kitchen | ● | □ | ⇨ | D | ▽ |
| 20.10 | Parallelism | PREPARE THE PACKAGING | | | | | |
| 20.10.10 | Get Paper | | ○ | □ | ➡ | D | ▼ |
| 20.10 | Parallelism | PARALLELISM END | | | | | |
| 20.10.10 | Get Box | | ○ | □ | ➡ | D | ▼ |
| 20.10 | Parallelism | PARALLELISM END | | | | | |
| 20.20 | Place Paper in Box | | ● | □ | ⇨ | D | ▽ |
| 20 | Parallelism | PARALLELISM END | | | | | |
| 30 | Put Muffins | | ● | □ | ⇨ | D | ▽ |
| 40 | End | | ● | □ | ⇨ | D | ▽ |
| **Author** | | | **Date** | | | | |
| Marko Vjestica | | | May 30, 2021 | | | | |

Figure 5. Example of generated ASME FPC

production process model can be done by using MultiProLan. Its models can be automatically transformed into different forms of technical documentation.

The proposed approach means that the data integrated in a model can easily be fed to series of technical documentation generators, each one being designed to generate a specific form of technical documentation. Therefore, maintenance and generation of various kinds of technical documentation can be customized and facilitated. This approach with the extended MultiProLan language provides the following benefits: easier modifications of a document template when required by legal regulations, standards or market changes; increased document consistency and quality; and minimized number of human errors due to documents being automatically generated in a non-stochastic manner. Also, this approach would significantly simplify the process of keeping the documents up to date, since all changes are only applied to the single point of knowledge instead of being spread over multiple documents.

In future work, generators for documents like GBOM, single-object flow charts and operation sheets can be created. Additionally, a graphic editor for easy and efficient customization of technical documentation templates by end users can be created. An evaluation is required to compare the efficiency of the proposed method with the current method of creating production process documentation. In future, document version control should also be integrated with the proposed approach to an enable efficient way to track and manage multiple versions of the documents. The focus of this paper was on generating technical documentation based only on production process models. To generate additional technical documentation that is related to a production system, another language is needed to create such models. As MultiProLan's syntax is tailored for the production process modeling, required extensions would be rather large and impractical. A similar approach still can be used for the generation of production system documentation, but it would require another modeling language tailored for the production system, as well as new generators.

## ACKNOWLEDGMENT

This paper is supported by KEBA AG Linz and by the Ministry of Education, Science and Technological Development through the project no. 451-03-68/2020-14/200156: "Innovative scientific and artistic research from the FTS domain".

## REFERENCES

[1] J. Jiao, M. M. Tseng, Q. Ma, and Y. Zou, "Generic Bill-of-Materials-and-Operations for High-Variety Production Management," *Concurr. Eng.*, vol. 8, no. 4, pp. 297–321, Dec. 2000, doi: 10.1177/1063293X0000800404.

[2] EUR-Lex, "EUR-Lex - 02011L0065-20140129 - EN," Directive, 2011. Accessed: May 25, 2021. [Online]. Available: https://eur-lex.europa.eu/eli/dir/2011/65/2014-01-29

[3] The Office of Environmental Health Hazard Assessment (OEHHA), "Proposition 65 - Safe Drinking Water and Toxic Enforcement Act of 1986," California, Law, 1986. Accessed: May 26, 2021. [Online]. Available: https://oehha.ca.gov/proposition-65

[4] International Organization for Standardization (ISO), "ICS 01.110 - Technical product documentation," Standards. Accessed: May 26, 2021. [Online]. Available: https://www.iso.org/ics/01.110/x/

[5] American Society of Mechanical Engineers, *ASME Standard: Operation and Flow Process Charts*. New York: ASME, 1947.

[6] D. A. Zakoldaev, A. V. Gurjanov, A. V. Shukalov, and I. O. Zharinov, "The life cycle of technical documentation in the smart factory of Industry 4.0," in *IOP Conference Series: Materials Science and Engineering*, Russian Federation, 2019, vol. 665, p. 012016. doi: 10.1088/1757-899X/665/1/012016.

[7] A. Barthelmey, D. Störkle, B. Kuhlenkötter, and J. Deuse, "Cyber Physical Systems for Life Cycle Continuous Technical Documentation of Manufacturing Facilities," in *Proceedings of the 47th CIRP Conference on Manufacturing Systems*, 2014, vol. 17, pp. 207–211. doi: 10.1016/j.procir.2014.01.050.

[8] M. Dassisti *et al.*, "Industry 4.0 paradigm: The viewpoint of the small and medium enterprises," in *Proceedings of the 7th International Conference on Information Society and Technology (ICIST 2017)*, Kopaonik, Serbia, 2017, vol. 1, pp. 50–54.

[9] T. H.-J. Uhlemann, C. Lehmann, and R. Steinhilper, "The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0," in *Procedia CIRP*, 2017, vol. 61, pp. 335–340. doi: 10.1016/j.procir.2016.11.152.

[10] M. Pisarić, V. Dimitrieski, M. Vještica, and G. Krajoski, "Towards a Non-Disruptive System for Dynamic Orchestration of the Shop Floor," in *IFIP Advances in Information and Communication Technology (AICT)*, Novi Sad, Serbia, 2020, vol. 592, pp. 469–476. doi: 10.1007/978-3-030-57997-5_54.

[11] M. Vještica, V. Dimitrieski, M. Pisarić, S. Kordić, S. Ristić, and I. Luković, "The Syntax of a Multi-Level Production Process Modeling Language," in *Proceedings of the 2020 Federated Conference on Computer Science and Information Systems (FedCSIS 2020)*, Sofia, Bulgaria, Sep. 2020, vol. 21, pp. 751–760. doi: 10.15439/2020F176.

[12] K. Lenkenhoff, A. Barthelmey, K. Lemmerz, B. Kuhlenkötter, and J. Deuse, "Communication Architecture for Automatic Plant Documentation Updates," in *Procedia CIRP*, 2016, vol. 44, pp. 365–370. doi: 10.1016/j.procir.2016.02.100.

[13] R. Drath, A. Luder, J. Peschke, and L. Hundt, "AutomationML - the glue for seamless automation engineering," in *Proceedings of the 2008 IEEE International Conference on Emerging Technologies and Factory Automation*, Hamburg, Germany, 2008, pp. 616–623. doi: 10.1109/ETFA.2008.4638461.

[14] B. Weißenberger, S. Flad, X. Chen, S. Rösch, T. Voigt, and B. Vogel-Heuser, "Model driven engineering of manufacturing execution systems using a formal specification," in *Proceedings of 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, Luxembourg, Sep. 2015, pp. 1–8. doi: 10.1109/ETFA.2015.7301430.

[15] M. Witsch and B. Vogel-Heuser, "Towards a Formal Specification Framework for Manufacturing Execution Systems," *IEEE Trans. Ind. Inform.*, vol. 8, no. 2, pp. 311–320, May 2012, doi: 10.1109/TII.2012.2186585.

[16] Object Management Group, "Business Process Model and Notation, Version 2.0.2," Technical Report, 2014.

[17] Object Management Group, "Unified Modeling Language, Version 2.5.1," Technical Report, Dec. 2017.

[18] W. Reisig, *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*, 1st ed. Berlin Heidelberg: Springer-Verlag, 2013. doi: 10.1007/978-3-642-33278-4.

[19] M. Vještica, V. Dimitrieski, M. Pisarić, S. Kordić, S. Ristić, and I. Luković, "Towards a Formal Specification of Production Processes Suitable for Automatic Execution," *Open Comput. Sci.*, vol. 11, no. 1, pp. 161–179, 2021, doi: 10.1515/comp-2020-0200.

[20] M. Vještica, V. Dimitrieski, M. Pisarić, S. Kordić, S. Ristić, and I. Luković, "An Application of a DSML in Industry 4.0 Production Processes," in *IFIP Advances in Information and Communication Technology (AICT)*, Novi Sad, Serbia, 2020, vol. 591, pp. 441–448. doi: 10.1007/978-3-030-57993-7_50.

[21] D. Dinero, *Training Within Industry: The Foundation of Lean*, 1st ed. CRC Press, Taylor & Francis Group, 2005.

[22] D. H. Stamatis, *Failure Mode and Effect Analysis: FMEA from Theory to Execution*, 2nd ed. Milwaukee, Wisconsin, United States: American Society for Quality, Quality Press, 2003.

[23] G. Salvendy, Ed., *Handbook of Industrial Engineering: Technology and Operations Management*, 3rd ed. John Wiley & Sons, Inc., 2001. doi: 10.1002/9780470172339.